# Programming Assignment 1
# Supply Chain Management System – Blockchain Technology

Imagine you are working in a manufacturing company. It manufactures more than ten products and is currently facing some issues in its distribution to clients. The issues are listed later on. You are expected to make a supply chain management system which checks that the product is reaching the clients from the manufacturer with the help of distributors.

**To prevent errors in the distribution of products, you need to build a supply chain management system using blockchain technology with the following features:**

1. To register new clients, distributors and a manufacturer (only one) to the system, with the client and distributor depositing a security amount to a trusted third party.
2. To improve the security of the blockchain, incorporate a **consensus algorithm** assigned to your group.
3. Implementing **Merkle tree** to calculate the hash of all the transactions in a block and successfully mine the block with the transaction.
4. To view the current product status in the supply chain using a **QR code**.
5. At one time, the distributor can distribute a product to a dedicated client. Once the transaction is confirmed by both the distributor and the consumer, then only the next delivery can be taken by him/her.
6. A well-known issue is understood when:
    a. The distributor has dispatched the product, and the client has received it, but the client is denying it (The client is lying, but the distributor is not).
    b. The distributor has not dispatched the product, and the client has not received it (The client is not lying, but the distributor is).

Resolve both issues & after identifying the liar, make deductions from the security deposit if the distributor or consumer is telling a lie.

| Distributor Say | In Real | Consumer Say | In Real |
|---|---|---|---|
| Product Dispatched | Product Dispatched | Product NOT Received | Product Received |
| Product Dispatched | Product NOT Dispatched | Product NOT Received | Product NOT Received |

Hint: To check these conditions, you can use unique Product IDs and keys.

## The different consensus algorithms that you will be implementing are:

1) Proof Of Stake (PoS)
2) Proof of Elapsed Time (PoET)
3) Delegated Proof-of-Stake (DPoS)
4) Proof of Work (PoW)

The consensus algorithm which you will be implementing will be randomly assigned.

## Basic Structure of a transaction that is expected:

Manufacture ID/name, Distributor ID/name, Client ID/name, Amount(optional), Timestamp of the transaction.

**{**
**Distributor (Di) got from the manufacturer -> Timestamp 1**
**Distributor dispatched ->Timestamp 2**
**Client (Ci) received -> Timestamp 3**
**}**

## Basic Structure of a Block in the blockchain:

Timestamp, Merkle root, Hash of the previous block, etc

Note: Above mentioned is the basic structure that is expected from you, but you are free to improvise.

Implement the blockchain using *Python/Go/Java/Javascript* at your convenience. *Choose a language you are familiar with, and you are free to use existing inbuilt libraries of the chosen language.*

For example, in Python, the Hash of blocks can be calculated using the “hashlib” library.

Note: *You are not expected to use a Database management system* and do any API calls (API calls can be done using Chainlink). Doing API calls is optional and may not be needed.

## Marking Scheme:

| | |
|---|---|
| Feature 1 | 1 mark |
| Feature 2 | 2 marks |
| Feature 3 | 2 marks |
| Feature 4 | 3 marks |
| Feature 5 | 1 mark |
| Feature 6 | 3 marks |

## General Instructions:

1) Hints are only meant to get you started; your innovation and extra features are welcome.
2) Make sure you only implement the algorithm assigned to you as part of this assignment you will not be awarded any marks for the feature.
3) The code should be readable and adequately commented.
4) Since there are a lot of files in most of your previous submissions, make sure you mention the part where you implemented the algorithm, failing which you will not be rewarded for the same.
5) Make sure to include group information in the readme.
6) A detailed explanation of functions written in the code in the readme file

## Submission Guidelines:

1) Make a zip/rar file of the project and submit it to the Google form that will be shared with you later
2) Only one member of the team should be making the Assignment Submission.
3) Make sure to include a Readme file with the submission.
4) The name of the zip file will be A1_Group_No_xyz.zip (Ex: A1_Group_No_1.zip)
5) Make your submissions here : https://forms.gle/KjtwteFpcSD5yRM19
6) Submission Deadline :: **Sept 28th 11:59pm**
7) If you have any further queries, you can mail them to:
   Jinil Shah : f20201750@hyderabad.bits-pilani.ac.in
   Samandeep Singh : f20200065@hyderabad.bits-pilani.ac.in